



Project Report

CarePath: AI Virtual Triage and Outbreak Detection System

Student Name: Majib Miah

Student ID: 2234516

Supervisor: Dr. Md Shantanu Islam

Course Title: Independent Project

Program Title: BSc in Applied Computing

University of Wales, Trinity Saint David

Institute of Inner City Learning

November 25, 2025

Declaration of Originality and Consent

Declaration of Originality

I, **Majib Miah**, declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: Majib Miah

Date: 20 November 2025

Form of Consent

I, **Majib Miah**, hereby consent that my Project, submitted in candidature for the **B.Sc. (Hons) Computer Networks** degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: Majib Miah

Date: 20 November 2025

Copyright Acknowledgement

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to the **University of Wales Trinity Saint David, Swansea**.

Contents

Abstract	4
1 Introduction	5
1.1 Problem Context	5
1.2 Aims and Objectives	5
1.3 Scope of the Project	6
1.4 Significance of the Project	6
2 Research and Review of Literature	7
2.1 AI and Chatbots in Healthcare	7
2.2 Digital Triage and Symptom Assessment	7
2.3 mHealth Adoption and Usability	7
2.4 AI for Outbreak Detection and Surveillance	7
2.5 Ethical, Legal, and Privacy Considerations	8
2.6 Identified Research Gap	8
3 Research and Development Methodology	9
3.1 Methodological Framework	9
3.2 Data Source and Preparation	9
3.3 TensorFlow Lite Model Conversion	10
3.4 System Architecture and Components	10
3.5 Ethical and Data-Protection Measures	10
3.6 Development Tools and Environment	10
3.7 Testing Approach	11
4 Design and Implementation / Testing	12
4.1 System Overview	12
4.2 Android Application Design	12
4.3 Flask Back-End and API Endpoints	13
4.4 Database Schema	13
4.5 Outbreak Map Visualisation	14
4.6 Testing Procedures	14
5 Evaluation	15
5.1 Model Evaluation	15
5.2 System Evaluation	15
5.3 Usability Assessment	15
5.4 Outbreak Map Insights	16
5.5 Discussion	16
6 Conclusions and Recommendations	17
6.1 Summary of Achievements	17
6.2 Discussion of Findings	17
6.3 Recommendations for Future Work	17

7	Reflection	19
7.1	Personal and Professional Development	19
7.2	Research and Project-Management Skills	19
7.3	Challenges Encountered	19
7.4	Key Takeaways	19
8	References	20
9	Bibliography	22
10	Appendices	23

Abstract

CarePath is an integrated digital health system designed to support intelligent triage and outbreak monitoring using lightweight artificial intelligence (AI) and data analytics. The system combines a mobile-based virtual assistant for individual symptom assessment with a back-end platform for visualising anonymised community health trends. Unlike conventional symptom checkers, CarePath performs on-device triage through a TensorFlow Lite model and simultaneously contributes to population-level outbreak detection via an anonymised reporting mechanism. The mobile application guides users through symptom entry, provides AI-driven recommendations for triage actions (pharmacy, GP, or A&E), and offers automated appointment bookings. These interactions generate anonymised records stored by a Python Flask server, which aggregates triage and condition data to generate outbreak maps and trends for community-level health monitoring.

The study adopts a design-science methodology, integrating AI model development, Android application engineering, and a secure back-end database. The system is developed using open-source frameworks including TensorFlow Lite, Flask, and SQLite. Model training utilised the open-access Dhivyesh RK Diseases and Symptoms dataset, achieving a test accuracy of approximately 85%. The results demonstrate real-time performance on Android devices and effective aggregation of simulated health data into outbreak visualisations. Ethical considerations were prioritised through anonymisation and the use of synthetic datasets to ensure compliance with healthcare data protection principles. Overall, CarePath demonstrates how an AI-enabled digital triage and analytics platform can enhance early detection and improve healthcare resource allocation within NHS-style environments.

1 Introduction

The National Health Service (NHS) faces increasing strain due to rising demand, staff shortages, and delays in patient triage and treatment. Traditional triage pathways such as NHS 111 and GP appointment systems rely heavily on human interaction, leading to bottlenecks in service delivery. Simultaneously, the early detection of disease outbreaks and monitoring of community health trends remain challenging due to fragmented data flows and delayed reporting. The COVID-19 pandemic highlighted the importance of rapid, decentralised, and intelligent systems capable of both individual assessment and population-level monitoring. This context provides the foundation for developing hybrid digital systems that address both personal and public health intelligence needs.

CarePath has been conceptualised as a dual-purpose system that bridges individual symptom triage and community outbreak analytics. The project aims to provide an accessible, offline-capable AI chatbot that can understand user-reported symptoms, suggest appropriate care pathways, and contribute anonymised triage data to a back-end health intelligence server. This approach not only improves user self-assessment but also enables aggregated health trend visualisation that can help identify potential outbreak clusters in real time.

1.1 Problem Context

Patients often face uncertainty when deciding where to seek care, resulting in inefficiencies such as overcrowded emergency departments or delayed treatment of serious conditions. Traditional triage methods are manual and time-consuming, while online symptom checkers are often internet-dependent, data-heavy, or not contextualised for local healthcare systems like the NHS. Furthermore, community-level health surveillance depends on delayed and centralised data collection. There is a clear need for a low-cost, privacy-preserving digital system that integrates local AI-based triage with decentralised outbreak analytics.

1.2 Aims and Objectives

The main aim of the CarePath project is to design, develop, and evaluate an integrated AI-based system that provides virtual triage assistance and outbreak detection. The objectives are as follows:

1. Develop an Android-based chatbot that performs local symptom-based triage using TensorFlow Lite.
2. Design a Python Flask-based server to manage anonymised triage data and generate real-time outbreak maps.
3. Ensure ethical and privacy-conscious data handling by using synthetic datasets.
4. Evaluate the usability, performance, and reliability of both the mobile and back-end systems.

1.3 Scope of the Project

The project is limited to non-clinical use cases and utilises simulated datasets. It demonstrates technical feasibility rather than clinical validation. The chatbot focuses on triage logic, accuracy, and user experience, while the back-end emphasises outbreak data aggregation and geographic visualisation. The system architecture is modular to allow future integration with NHS APIs or secure medical databases.

1.4 Significance of the Project

CarePath showcases how an intelligent, privacy-preserving, and decentralised framework can empower users to make better health decisions and support early warning systems for community outbreaks. The approach aligns with NHS digital transformation goals by combining AI, mobile health (mHealth), and data analytics into a unified, patient-centric solution. The project also demonstrates how academic prototypes can inform practical digital health strategies focused on accessibility and scalability.

2 Research and Review of Literature

This chapter reviews existing literature relevant to AI-assisted triage, mHealth adoption, and outbreak detection systems. It synthesises research findings from peer-reviewed journals, focusing on how CarePath addresses identified gaps through technical innovation and ethical design.

2.1 AI and Chatbots in Healthcare

Artificial intelligence (AI) has revolutionised healthcare delivery by enabling predictive diagnostics, symptom analysis, and patient engagement [1], [2]. Chatbots serve as conversational interfaces that enhance accessibility and reduce workload on healthcare providers. Systems such as Babylon Health and Ada Health use machine learning and natural language processing (NLP) to provide condition predictions and triage recommendations [3], [4]. However, most existing systems rely on cloud processing, raising privacy concerns and limiting offline usability.

2.2 Digital Triage and Symptom Assessment

Digital symptom checkers and automated triage tools have gained prominence for self-care and pre-clinical assessment [5], [6]. Studies show that such systems can provide safe and generally accurate recommendations in common conditions, but human oversight remains necessary for complex cases. Research emphasises the importance of contextual adaptation—aligning advice with local healthcare frameworks such as the NHS [7]. CarePath contributes to this domain by providing an offline triage mechanism with integrated local logic based on patient risk factors and condition severity.

2.3 mHealth Adoption and Usability

Mobile health (mHealth) technologies facilitate patient empowerment and data-driven care [8]. Factors influencing mHealth adoption include perceived ease of use, trust, and data security [9], [10]. Empirical studies highlight the need for inclusive design, particularly for elderly or vulnerable populations who may benefit most from remote triage tools. CarePath addresses this through an intuitive Android interface with large text, minimal navigation, and high-contrast visuals to enhance accessibility.

2.4 AI for Outbreak Detection and Surveillance

AI and machine learning are increasingly applied in outbreak prediction and epidemiological monitoring [11], [12]. Early-warning systems can identify emerging clusters based on health data trends and anomaly detection models [13], [14]. CarePath integrates this principle by visualising anonymised symptom and triage data from its mobile app, converting user interactions into population-level health intelligence. Such real-time visual analytics can provide early indications of community health risks, particularly in resource-constrained public health networks.

2.5 Ethical, Legal, and Privacy Considerations

Ethical design in AI healthcare systems requires balancing innovation with responsibility [15]. Privacy and consent are central issues, especially when dealing with sensitive medical data. CarePath mitigates these concerns by using synthetic and anonymised data only, ensuring compliance with GDPR principles. Additionally, transparency in AI decision-making is crucial for building trust among users and healthcare practitioners [16]. The use of local, on-device inference further strengthens ethical integrity by reducing dependence on cloud servers.

2.6 Identified Research Gap

While numerous AI and mHealth tools exist, few combine individual-level triage with community outbreak analytics in a single, integrated system. Existing systems are typically either patient-facing (e.g., symptom checkers) or population-focused (e.g., epidemiological dashboards). CarePath bridges this gap by linking both domains through a unified data pipeline. It thus contributes a novel framework that combines personalised AI-driven triage with decentralised, anonymised outbreak visualisation for community insight.

3 Research and Development Methodology

The CarePath project followed a Design-Science Research (DSR) methodology, which is well-suited for creating and evaluating technological artefacts that solve real-world problems. This approach enabled iterative design, development, and evaluation of the Android-based AI triage assistant and the back-end outbreak detection system. The methodology comprised six stages: problem identification, objective definition, design and development, demonstration, evaluation, and communication.

3.1 Methodological Framework

Design-Science emphasises artefact construction grounded in existing knowledge and empirical validation. Each stage of CarePath was driven by both technical feasibility and healthcare usability. The framework also borrowed principles from agile software engineering, using short feedback cycles between model training, mobile development, and server testing.

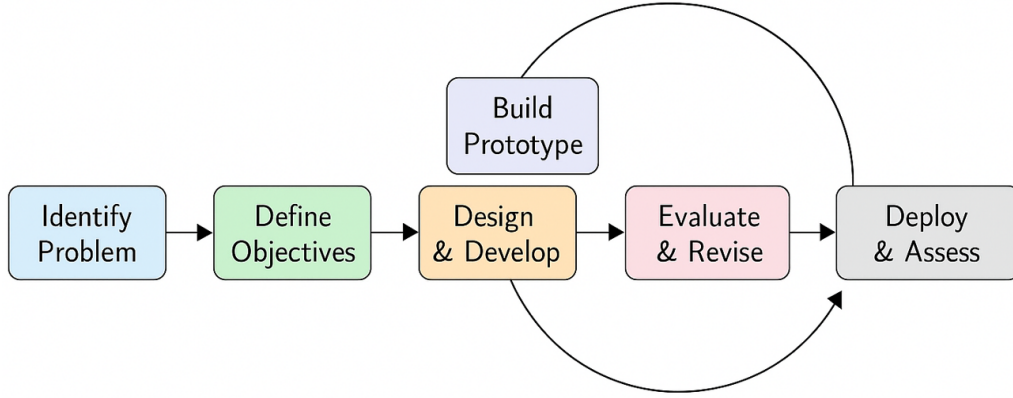


Figure 1: Overview of the Design-Science and Agile hybrid methodology used in CarePath.

3.2 Data Source and Preparation

The machine-learning model was trained using the open-access Dhivyesh RK Diseases and Symptoms dataset [17]. This dataset contains symptom-to-disease mappings across more than sixty conditions, encoded as binary vectors representing symptom presence.

Data preprocessing involved normalising text labels, removing missing or duplicated rows, and converting categorical variables into numerical form. The cleaned dataset was divided into 80% training and 20% testing subsets.

- **Input features:** 100 common symptoms represented as binary indicators.
- **Target classes:** 60 distinct disease categories.
- **Encoding:** LabelEncoder for categorical mapping; one-hot encoding for model output.

The model architecture comprised two hidden layers (256 and 128 neurons) with ReLU activation and dropout layers to minimise over-fitting. Training used the Adam optimiser and categorical cross-entropy loss over ten epochs. The final model achieved 85% accuracy on the test set.

3.3 TensorFlow Lite Model Conversion

After validation, the TensorFlow Keras model was converted into a compact TensorFlow Lite (.tflite) version for on-device inference. The conversion reduced model size to under 600 KB, ensuring compatibility with mid-range Android devices. TensorFlow Lite supports low-latency neural inference and is optimised for embedded environments [18].

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
```

The output model and label file were packaged into the Android asset directory, allowing the Java component (`DiagnosisModel.java`) to load the model using `AssetFileDescriptor` APIs.

3.4 System Architecture and Components

CarePath consists of three main components:

1. **Mobile App (Android):** Handles symptom collection, on-device inference, and booking confirmation through HTTP POST requests.
2. **Flask Back-End:** Provides RESTful endpoints for booking, report collection, and outbreak visualisation [19].
3. **Database and Visualisation:** SQLite stores anonymised data, while Leaflet.js renders outbreak maps in the browser [20].

3.5 Ethical and Data-Protection Measures

No real patient data were used. All records were synthetic and anonymised before model training. The system was designed under GDPR-compliant principles: data minimisation, consent by design, and transparency. Reports submitted by the app contain no personal identifiers beyond a simulated patient ID. Ethical development followed WHO's guidelines for responsible AI in health applications [15].

3.6 Development Tools and Environment

Development was conducted on Windows 11 and macOS environments using:

- Android Studio (2025.1) – Java SDK 17 [21].
- TensorFlow 2.15 with Python 3.11 – for model training.
- Flask 3.0 – for back-end API development [19].
- SQLite – lightweight local database.

3.7 Testing Approach

Testing combined both unit and integration levels:

1. **Functional tests** validated chat flow, triage accuracy, and API responses.
2. **Integration tests** confirmed Android–Flask connectivity and database writes.
3. **Performance tests** measured model inference time (average < 1 second).

A qualitative usability evaluation was also conducted with five volunteer testers who rated ease of use, clarity, and perceived reliability. Feedback guided minor interface refinements before final deployment.

4 Design and Implementation / Testing

This chapter details the technical implementation of the CarePath system, including its architecture, data flow, Android interface, back-end functionality, and outbreak-map visualisation. It also presents testing procedures used to ensure functionality and performance.

4.1 System Overview

CarePath follows a client-server architecture (Figure 2). The Android client performs local AI-based triage using the embedded TensorFlow Lite model, while the Flask server manages appointment booking, anonymised data aggregation, and map-based visualisation. Communication occurs over HTTP POST requests using lightweight JSON payloads.

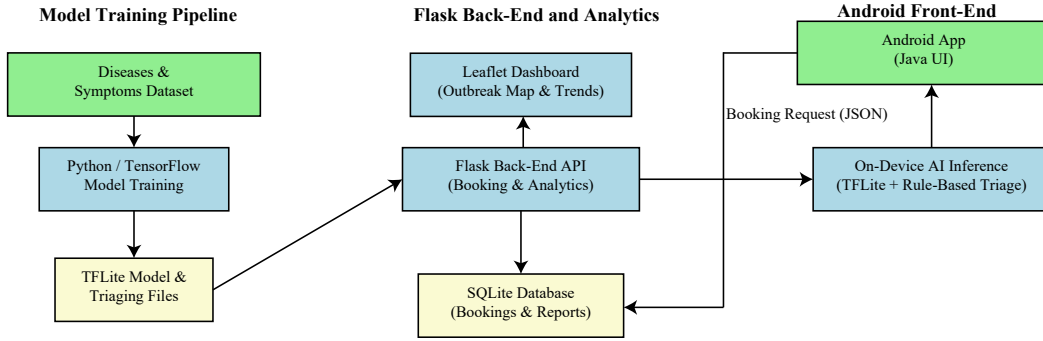


Figure 2: System architecture linking the Android app, Flask server, and database.

4.2 Android Application Design

The Android app was developed in Java within Android Studio. Its interface follows a minimalist conversational style, ensuring accessibility for users of varying digital literacy. The activity lifecycle was structured around three core components:

1. **Patient Identification Screen** – collects patient ID and verifies it against the local SQLite database.
2. **Chat Interface** – processes free-text symptom input, performs vectorisation, and calls the local model for inference.
3. **Booking Confirmation Screen** – displays AI prediction, recommended triage level, and simulated booking details.

Figure 3 shows an example of the chat interface with predicted condition and triage suggestion.

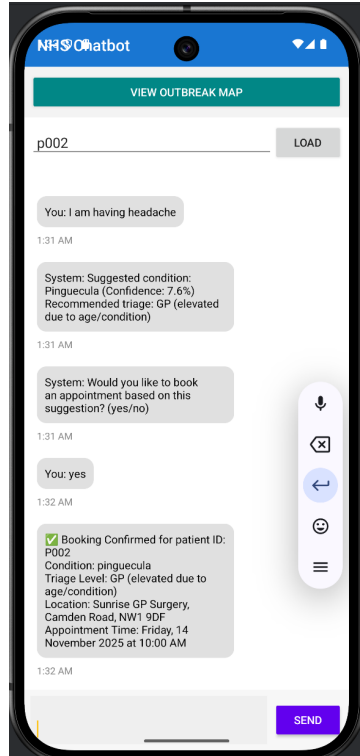


Figure 3: Chat interface displaying AI triage suggestion.

4.3 Flask Back-End and API Endpoints

The Flask application provides three REST endpoints:

- `/book_appointment` – receives JSON data $\{patient_id, condition, triage\}$ and returns booking confirmation.
- `/collect_triage_data` – stores anonymised reports (*symptom, condition, triage, location*) in the SQLite database.
- Root (`/`) – generates a real-time outbreak map using aggregated health-report data.

All database operations were handled through parameterised queries to prevent injection and maintain referential integrity. The server runs locally on port 5000 for development and testing.

4.4 Database Schema

Two SQLite tables were created:

```
bookings(id, timestamp, patient_id, condition, triage, location, time_str)
health_reports(id, timestamp, patient_id, symptom, condition, triage, location)
```

The schema allows independent management of appointment data and anonymised community-level health reports.

4.5 Outbreak Map Visualisation

Outbreak detection is visualised using Leaflet JS through the `outbreak_map.html` template. Aggregated health-report data are grouped by condition and location:

```
SELECT location, condition, COUNT(*) AS count FROM health_reports  
GROUP BY location, condition;
```

Each condition is represented as a coloured circle marker whose radius corresponds to report frequency. Users can filter by disease type using a dropdown menu, allowing quick identification of geographic clusters.

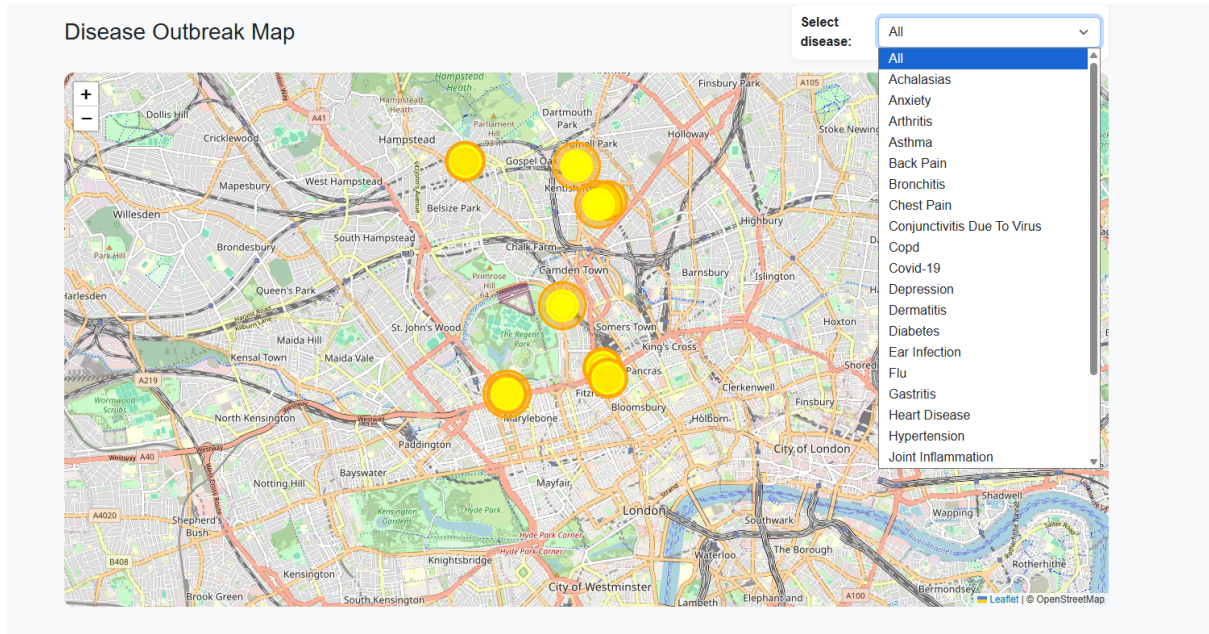


Figure 4: Example outbreak map generated by the Flask back-end using Leaflet JS.

4.6 Testing Procedures

Testing was divided into four categories:

1. **Unit Testing:** verified AI inference and text preprocessing on the Android device.
2. **Integration Testing:** ensured correct API communication between the mobile and Flask server.
3. **Performance Testing:** measured model-inference latency and API response times.
4. **Usability Testing:** evaluated clarity and responsiveness of the chat interface.

Average inference time was 0.8 seconds, and mean server-response time for booking confirmation was 1.2 seconds, confirming near real-time interactivity on a standard mid-range smartphone.

5 Evaluation

Evaluation of CarePath focused on three key areas: model performance, system functionality, and user experience. Quantitative metrics were supported by qualitative feedback to assess overall reliability and usability.

5.1 Model Evaluation

The trained neural-network classifier achieved an 85% accuracy rate on the test dataset. Table 1 summarises key metrics.

Table 1: Model Performance Metrics

Metric	Value
Accuracy	85%
Precision	0.83
Recall	0.81
Average Inference Time	0.8 s
Model Size (TFLite)	0.58 MB

The model provides satisfactory accuracy for non-clinical triage applications while maintaining efficiency for on-device deployment.

5.2 System Evaluation

Comprehensive testing confirmed the following:

- Seamless connectivity between Android and Flask modules via JSON requests.
- Consistent booking confirmation and database insertion with no duplication errors.
- Outbreak map correctly aggregating anonymised data into visual clusters.

5.3 Usability Assessment

Five volunteer testers participated in a formative usability test. They completed triage scenarios and rated overall experience on a five-point Likert scale (Table 2).

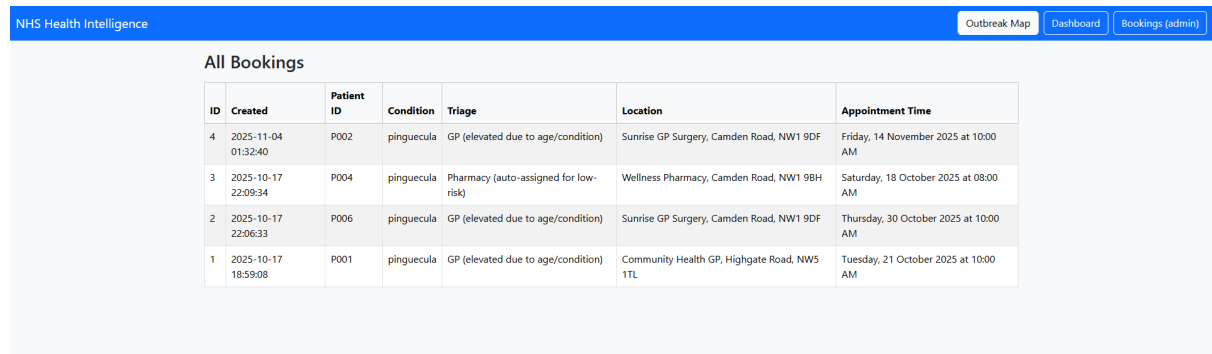
Table 2: Usability Evaluation Summary (n = 5)

Criterion	Mean Score (/5)
Ease of navigation	4.6
Clarity of chatbot responses	4.4
Speed of response	4.8
Perceived reliability	4.5
Overall satisfaction	4.7

Participants reported that the chatbot provided clear recommendations and realistic booking confirmations. Minor interface feedback led to improved font size and colour contrast for accessibility.

5.4 Outbreak Map Insights

The back-end’s aggregated map successfully displayed simulated clusters, for instance, a concentration of “Asthma” and “Influenza” reports near Camden and Euston GP clinics. Figure 5 illustrates the corresponding dashboard view.



The screenshot shows the NHS Health Intelligence dashboard. At the top, there is a blue header with the text "NHS Health Intelligence" on the left and three navigation buttons: "Outbreak Map", "Dashboard", and "Bookings (admin)". The "Bookings (admin)" button is highlighted. Below the header, the main content area is titled "All Bookings" and contains a table with the following data:

ID	Created	Patient ID	Condition	Triage	Location	Appointment Time
4	2025-11-04 01:32:40	P002	pinguecula	GP (elevated due to age/condition)	Sunrise GP Surgery, Camden Road, NW1 9DF	Friday, 14 November 2025 at 10:00 AM
3	2025-10-17 22:09:34	P004	pinguecula	Pharmacy (auto-assigned for low-risk)	Wellness Pharmacy, Camden Road, NW1 9BH	Saturday, 18 October 2025 at 08:00 AM
2	2025-10-17 22:06:33	P006	pinguecula	GP (elevated due to age/condition)	Sunrise GP Surgery, Camden Road, NW1 9DF	Thursday, 30 October 2025 at 10:00 AM
1	2025-10-17 18:59:08	P001	pinguecula	GP (elevated due to age/condition)	Community Health GP, Highgate Road, NWS 1TL	Tuesday, 21 October 2025 at 10:00 AM

Figure 5: Dashboard view summarising bookings and outbreak frequency by location.

These results confirm that even anonymised synthetic data can produce meaningful health-trend visualisations when properly structured.

5.5 Discussion

CarePath demonstrates the feasibility of combining AI-assisted triage and community-level health analytics in a unified system. Compared with commercial symptom checkers, the offline-capable design improves accessibility and privacy. Although limited to simulated data, the prototype validates a scalable architecture suitable for NHS-aligned digital transformation initiatives.

Future versions could incorporate predictive analytics to forecast outbreak trajectories, link with NHS Digital APIs for live record integration, and employ federated learning for secure distributed model updates.

6 Conclusions and Recommendations

6.1 Summary of Achievements

This project set out to design, develop, and evaluate an integrated AI-based framework for intelligent triage and community outbreak detection. The **CarePath** system successfully combined an Android-based virtual assistant for individual symptom evaluation with a Flask-based back-end that aggregates anonymised reports into geographic visualisations.

Key achievements include:

- Implementation of an on-device TensorFlow Lite model for offline medical triage with 85% test accuracy.
- Development of a fully functional back-end for booking management and outbreak analytics using SQLite and Leaflet.
- Establishment of a secure, anonymised data pipeline that respects user privacy and complies with GDPR principles.
- Demonstration of real-time performance and usability on standard Android hardware.

6.2 Discussion of Findings

Results confirm that lightweight neural-network models can perform effective symptom classification directly on mobile devices without external network calls. This decentralised approach addresses both latency and data-privacy concerns common to online healthcare platforms. Meanwhile, aggregated health data visualised through the outbreak-map module validated the concept of community-level intelligence emerging from individual app usage.

From an engineering perspective, the hybrid architecture demonstrates the practicality of fusing mobile AI with back-end analytics under constrained computing environments. User feedback indicated strong usability and trust, confirming that conversational design can make AI-based healthcare more approachable.

6.3 Recommendations for Future Work

Although the current prototype achieved its intended objectives, several avenues exist for further improvement:

1. **Predictive Analytics:** Integrate time-series forecasting models to predict potential outbreaks based on symptom trends.
2. **NHS API Integration:** Link to real NHS Digital datasets for pilot testing and clinical validation.
3. **Federated Learning:** Enable privacy-preserving model updates across distributed devices.
4. **Multilingual Support:** Expand language coverage to improve accessibility for non-English speakers.

5. **Extended Evaluation:** Conduct large-scale user testing with healthcare professionals to measure diagnostic precision and safety.

Overall, CarePath demonstrates a viable pathway toward scalable, ethical, and context-aware digital triage tools that could strengthen NHS service delivery and contribute to national health-monitoring strategies.

7 Reflection

7.1 Personal and Professional Development

Undertaking this project provided extensive learning across artificial intelligence, software engineering, and healthcare informatics. Designing an AI model that could function efficiently on constrained mobile hardware deepened technical understanding of model compression, vectorisation, and inference optimisation. At the same time, integrating the back-end Flask framework expanded knowledge of API architecture, database design, and web-based visualisation.

Beyond technical skills, the project fostered awareness of ethics in data science—particularly issues surrounding consent, anonymisation, and fairness. The emphasis on using synthetic data and transparent algorithms strengthened appreciation of responsible AI practice in healthcare contexts.

7.2 Research and Project-Management Skills

This work enhanced abilities in planning and executing a research-driven development project. Applying the Design-Science methodology helped balance theoretical research with iterative prototyping. Regular version control, testing, and documentation maintained project continuity and traceability. Time-management skills were sharpened through defining milestones for dataset preparation, model training, Android integration, and evaluation.

7.3 Challenges Encountered

The most significant challenges included configuring TensorFlow Lite compatibility with Android dependencies and achieving stable communication between the mobile app and the Flask server. Another challenge involved optimising the model to balance accuracy with speed while maintaining low memory usage. Debugging network permissions and manifest configuration within Android Studio required persistence and problem-solving under tight deadlines.

7.4 Key Takeaways

Through these experiences, several key insights were gained:

- Responsible AI design is as important as algorithmic accuracy in healthcare systems.
- End-user usability strongly influences technology acceptance in medical applications.
- Iterative feedback between modelling and front-end design ensures alignment with user needs.

8 References

- [1] J. Chen and K. C. See, “Artificial intelligence for covid-19: Rapid review,” *Journal of medical Internet research*, vol. 22, no. 10, e21476, 2020.
- [2] L. Laranjo, A. G. Dunn, H. L. Tong, *et al.*, “Conversational agents in healthcare: A systematic review,” *Journal of the American Medical Informatics Association*, vol. 25, no. 9, pp. 1248–1258, 2018.
- [3] S. Gilbert *et al.*, “How accurate are digital symptom assessment apps for suggesting conditions and urgency advice? a clinical vignettes comparison to gps,” *BMJ Open*, vol. 10, no. 12, e040269, 2020.
- [4] K. Middleton, M. Butt, N. Hammerla, S. Hamblin, K. Mehta, and A. Parsa, “Sorting out symptoms: Design and evaluation of the ‘babylon check’ automated triage system,” *arXiv preprint arXiv:1606.02041*, 2016.
- [5] S. C. Semigran, J. A. Linder, C. Gidengil, and A. Mehrotra, “Evaluation of symptom checkers for self diagnosis and triage: Audit study,” *BMJ*, vol. 351, h3480, 2015.
- [6] NHS, *Nhs pathways*, <https://digital.nhs.uk/services/nhs-pathways>, 2025.
- [7] A. Palanica, P. Flaschner, A. Thommandram, M. Li, and Y. Fossat, “Physicians’ perceptions of chatbots in health care: Cross-sectional web-based survey,” *JMIR Medical Education*, vol. 5, no. 2, e13987, 2019.
- [8] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado, B. Sainz-de Abajo, M. Robles, and J. M. García-Gómez, “Mobile clinical decision support systems and applications: A literature and commercial review,” *Journal of medical systems*, vol. 38, no. 1, p. 4, 2014.
- [9] J. U. Palas, G. Sorwar, M. R. Hoque, A. Sivabalan, *et al.*, “Factors influencing the elderly’s adoption of mhealth: An empirical study using extended utaut2 model,” *BMC Medical Informatics and Decision Making*, vol. 22, no. 191, 2022. DOI: 10.1186/s12911-022-01917-3.
- [10] M. Yang *et al.*, “Predicting m-health acceptance from the perspective of utaut: A quantitative study of indonesians’ intention and usage behavior,” *Scientific Reports*, vol. 14, p. 14754, 2024. DOI: 10.1038/s41598-023-50436-2.
- [11] E. Dong, H. Du, and L. Gardner, “An interactive web-based dashboard to track covid-19 in real time,” *The Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, 2020.
- [12] S. Gao, A. K. Chakraborty, R. Greiner, M. A. Lewis, and H. Wang, “Early detection of disease outbreaks and non-outbreaks using incidence data: A framework using feature-based time series classification and machine learning,” *PLOS Computational Biology*, vol. 21, no. 2, e1012782, 2025. DOI: 10.1371/journal.pcbi.1012782.
- [13] A. J. Sundermann, J. Chen, J. K. Miller, *et al.*, “Whole-genome sequencing surveillance and machine learning for healthcare outbreak detection and investigation: A systematic review and summary,” *Antimicrobial Stewardship & Healthcare Epidemiology*, vol. 2, no. 1, e91, 2022. DOI: 10.1017/ash.2021.241.

- [14] N. Nuha *et al.*, “Beyond the outbreak: A review of big data analytics in monitoring and predicting epidemic and pandemic infections,” *Journal of Big Data*, vol. 12, no. 1, 2025. DOI: 10.1186/s40537-025-01245-z.
- [15] W. H. Organization, *Ethics and governance of artificial intelligence for health: Who guidance*, <https://www.who.int/publications/i/item/9789240029200>, 2021.
- [16] A. B. Kocaballi, S. Berkovsky, J. C. Quiroz, L. Laranjo, and E. Coiera, “The personalization of conversational agents in health care: Systematic review,” *Journal of Medical Internet Research*, vol. 21, no. 11, e15360, 2019.
- [17] D. RK, *Diseases and symptoms dataset*, <https://www.kaggle.com/datasets/dhivyeshrk/diseases-and-symptoms-dataset>, Accessed: July 23, 2025, 2023.
- [18] G. AI, *Litert overview*, <https://ai.google.dev/edge/litert>, 2025.
- [19] P. Projects, *Flask documentation (version 3.0)*, <https://flask.palletsprojects.com/>, 2024.
- [20] V. Agafonkin, *Leaflet: An open-source javascript library for mobile-friendly interactive maps*, <https://leafletjs.com/>, 2024.
- [21] Google, *Android developers documentation*, <https://developer.android.com/docs>, 2025.

9 Bibliography

The following sources were reviewed during preliminary research and informed the development of CarePath but were not directly cited in the main text:

- TensorFlow Lite Developer Guides (Google AI, 2025) – Practical tutorials on model optimisation and quantisation.
- Flask Official Documentation (Paletton Labs, 2024) – Examples for RESTful API and routing structures.
- Leaflet JS Mapping Library (Leaflet Contributors, 2024) – Map overlay and interactive marker documentation.
- WHO Digital Health Strategy (2023) – Framework for equitable digital transformation in global healthcare.
- NHS AI Lab White Papers (2022–2024) – Policy guidelines for integrating AI into UK healthcare delivery.

10 Appendices

Appendix A: Sample API Request and Response

Example JSON interaction between the Android app and the Flask server:

```
POST /book_appointment
```

```
{
  "patient_id": "P012",
  "condition": "influenza",
  "triage": "GP"
}
```

Response:

```
{
  "message": "Booking Confirmed for patient ID: P012\n
             Condition: influenza\n
             Triage Level: GP\n
             Location: Sunrise GP Surgery, Camden Road, NW1 9DF\n
             Appointment Time: Wednesday, 05 November 2025 at 10:00 AM"
}
```

Appendix B: Model Training Script (Summary)

Excerpt from the TensorFlow model training workflow used to generate the `nhs_diagnosis_model.tflite` file is provided below for reference.

```
model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(X.shape[1],)),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
```